

September 30, 2000

ICC Technical Program Committee Chair:

Dear Sir or Madam:

Please find attached my submission to ICC 2001. The paper is five pages long and formatted in two columns, as it would appear in the proceedings and as requested in the call for papers. In addition to the five pages, I included an abstract with the proofs of the theorems for the benefit of the reviewers. The abstract will be published in a technical report and not in the proceedings.

I am the sole author, and my contact information is:

Jorge A. Cobb  
Center for Advanced Telecommunications Systems and Services  
The University of Texas at Dallas  
Richardson, TX 75083-0688  
[jcobb@utdallas.edu](mailto:jcobb@utdallas.edu)  
phone: 972 883 2479  
fax: 972 883 2349

Thank you very much, and I look forward to hearing from you in December.

Sincerely yours,

Jorge A. Cobb  
Assistant Professor

# On Finding Optimum Paths in Computer Networks

Jorge A. Cobb<sup>1</sup>

Center for Advanced Telecommunications Systems and Services  
The University of Texas at Dallas  
Richardson, TX 75083-0688  
[jcobb@utdallas.edu](mailto:jcobb@utdallas.edu)

## Abstract

We consider the problem of finding an optimum path from each node in the network to a destination node in the network. Typically, each edge in the network is assigned a value, e.g. bandwidth or delay, and a metric function is applied consecutively to the edges in the path, e.g. minimum or maximum functions. The result is the path's metric. However, some metric functions have conflicts, and prevent all nodes from obtaining an optimum value simultaneously. Conditions to ensure that a metric function is conflict-free have been presented in the literature. In this study, we consider the more general problem where the optimality of the path is not a function of the edge weights but a function of the path itself. We therefore define a path ranking function, and give sufficient conditions for the ranking to be conflict free. Furthermore, a distributed algorithm is presented to obtain the optimum path from each node to the destination.

## 1. Introduction

One of the most significant factors affecting the performance of computer networks is the choice of routing algorithm. It is most significant, since its purpose is to find an optimum path between any pair of nodes in the network. Due to the importance of this algorithm, it must be chosen carefully, along with the metric that is used to measure the desirability of each path.

One of the fundamental principles of routing is known as the optimality principle [1][12], which states the following. Consider any pair of nodes  $u$  and  $w$  in the network, and an optimum path  $P$  from  $u$  to  $w$ . Then, if  $v$  is a node in  $P$ , then the subpath of  $P$  from  $v$  to  $w$  is also an optimum path from  $v$  to  $w$ . A consequence of this principle is the following:

- if each node independently chooses a neighbor along the optimum path to the destination, then all edges of the form (node, chosen neighbor to destination) form a spanning tree  $T$ , with the destination as its root.
- $T$  contains an optimum path from every node to the destination.

Hence, the above implies that there are no inconsistencies between the choice of path made by each node  $u$ , and the choice of path of any ancestors of  $u$  in  $T$ . Thus, a node

can be greedy, and always choose as its next hop to the destination the neighbor which yields the best path to the destination.

The wide range of traffic types to be supported by future networks has created a demand for a variety of routing metrics, which measure the desirability of a path. Thus, metrics may vary from minimum cost metrics [4] to maximum bottleneck bandwidth [3][5][13]. Thus, in addition to choosing a routing algorithm, the choice of routing metric is important for the applications. For a general discussion on a variety of routing metrics and algorithms, see [2].

However, while some metrics do satisfy the optimality principle, others do not. The minimum cost metric, where we desire a path whose sum of the cost of its edges is minimum, does satisfy the optimality principle (assuming all edge costs are positive and non-zero). On the other hand, as pointed out in [6], the maximum bandwidth metric, where we desire a path whose bottleneck bandwidth is the maximum, does not satisfy the optimality principle. This is illustrated in Figure 1. Here, one of the optimum paths from  $u$  to the destination  $r$  is  $(u, v, w, r)$ . However, none of the optimum paths from  $v$  to  $w$  contain  $(v, w, r)$ .

Furthermore, not only do some metrics not satisfy the optimality principle, their choice of paths to the destination may be conflicting. For example, assume that the metric is the pair (bandwidth, distance), where we desire the path with greatest bottleneck bandwidth, and whose distance is the smallest (a.k.a. "shortest widest path"). In Figure 1, there is a single optimum path from  $u$  to  $r$ , namely,  $(u, v, w, r)$ , and there is a single optimum path from  $v$  to  $r$ , namely,  $(v, x, z, r)$ . Thus, there is a conflict at node  $v$ .

In [6], necessary and sufficient properties were presented for these types of conflicts to be avoided, and hence, for a metric to define an optimum path for all nodes in the network. These properties may hold even though the optimality principle does not. For example, the maximum bandwidth metric does satisfy these properties, and thus is conflict free.

However, since the optimality principle does not hold, nodes cannot simply pick any neighbor along an optimal path as their next hop to a destination, since this may cause a loop. For example, an optimum path for  $x$  is  $(x, y, z, r)$ , and an optimum path for  $y$  is  $(y, x, z, r)$ . Thus, if  $x$  and  $y$  choose each other as the next hop, a loop is introduced. This may be solved by having each node choose as its next hop the neighbor along

---

<sup>1</sup> Supported in part by the Texas Advanced Research Program.

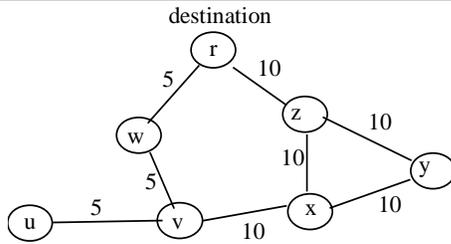


Figure 1: Routing for maximum bandwidth

the shortest optimum path. Thus, both  $x$  and  $y$  would choose  $z$  as their next hop.

The model of routing metrics discussed in [6] covers the usual metrics of assigning a value to each edge in the network, and the metric of a path consists on the consecutive application of a function (e.g. min or max) to the values of the edges in the path. This model, however, does not cover other metrics that may take into consideration the identities of the nodes along the path. This consideration is important in policy-based routing, where traffic is directed to or away from particular nodes in the network, irrespective of the quality of the edges along the path. The Border Gateway Protocol [10] used in the Internet is a particular example of policy-based routing.

In this paper, we consider routing metrics where the desirability of a path is a function that is not restricted to the weight of the edges in the path. Instead, we consider a function, which we call rank, whose input is a sequence of nodes (i.e. a path) and returns an integer, which expresses the desirability of the path.

Note that the use of a path rank function is more general than metrics based solely on edge weights. This is because a metric based on edge weights would assign the same metric to two different paths of equal length whose edge weights are the same. However, a path rank function may assign to them different ranks. Thus, path rank functions subsume edge metrics, and in consequence, some path rank functions will not satisfy the optimality principle.

Given that some path rank functions will not satisfy the optimality principle, the goal of this paper is to obtain sufficient conditions on the rank function such that a) no conflicts occur in the rank function, i.e., an optimum route can be obtained for every node in the network, and b) a simple distributed algorithm exists that allows each node to choose its next hop neighbor to the destination.

For readability and space restrictions, all proofs may be found in the appendix.

## 2. Path Selections

A network  $N$  is represented as an undirected graph  $(V, E)$ , where the node set  $V$  represents the set of computers in the network, and the edge set  $E$  represents the set of communication links in the network. For simplicity, we assume that  $N$  is connected. The results are easily extended when this is not the case.

A *path* in  $V$  is a sequence of nodes from  $V$ . A *network path*  $P$  in  $N$  is a sequence of nodes in  $V$  where each pair of consecutive nodes in  $P$  is an edge in  $E$ . We denote by

$head(P)$  the first node in the path of  $P$ .

We denote by  $P:Q$  the path obtained by concatenating paths  $P$  and  $Q$ . Also, we denote by  $P(v, w)$  the sub-path of  $P$  starting at node  $v$  and ending at node  $w$ .

Without loss of generality, we assume there is a single destination node in the network, which we call the *root* node. A path  $P$  in network  $N$  is *rooted* iff the last node in  $P$  is the root node, i.e.,  $P = v_k, v_{k-1}, \dots, v_0$  where  $v_0 = \text{root}$ .

Let  $v = head(P)$ , and assume node  $v$  has chosen  $P$  as its path to the root. Then, node  $v$  routes data messages to its neighbor  $w$  where  $w$  is the next node after  $v$  in  $P$ .

We assume each node  $v$  has a set of paths, denoted by  $Sel(v)$ , which are acceptable to  $v$ . That is, the path chosen from  $v$  to the root *must* be chosen from this set. In this section, we make no assumption as to how  $Sel(v)$  is chosen. In the next section, we define a specific set  $Sel(v)$  as the optimum paths with respect to a ranking function.

More formally, a *path selection* is a pair  $(V, Sel)$ , where:

- $V$  is a set of nodes, with a distinguished root node.
- $Sel()$  is a function, which takes a node  $v, v \in V - \{\text{root}\}$ , and returns a non-empty set of rooted paths over  $V$ , whose initial node is  $v$ .

Without loss of generality, we assume  $Sel(v) \neq \emptyset$  for all  $v$ .

If node  $v$  routes its data messages towards its neighbor  $w$ , then the remainder of the path which is followed by the data messages from  $v$  depends on the next hop choice made by  $w$ , and so on. This choice of next hop neighbors must be consistent to avoid undesirable conditions such as loops. We define this consistency formally as follows.

A rooted path  $P$  is *selectable* in  $(V, Sel)$  iff  $P \in Sel(v)$ , where  $v = head(P)$ . A path selection  $(V, Sel)$  is *satisfiable* in network  $N, N = (V, E)$ , iff there exists a spanning tree  $T$  of  $N$ , such that, for any  $v, v \in V$ , the rooted path from  $v$  to the root node along  $T$  is a selectable path in  $(V, Sel)$ . We refer to this spanning tree as a *satisfiability tree* of  $(V, Sel)$  in  $N$ .

Thus, each node  $v$  should choose as its next hop neighbor its parent in  $T$ . This guarantees that the path followed by the data messages of  $v$  is a selectable path. However, a path selection may not be satisfiable in some networks, and thus, it may not have a satisfiability spanning tree. Next, we present sufficient conditions to ensure the satisfiability of a path selection.

A path selection  $(V, Sel)$  is *consistent* iff it satisfies the following condition. For all  $v, w, P$ , and  $Q$ , where  $P \in Sel(v)$ ,  $w \in P$ , and  $Q \in Sel(w)$ ,

$$P(v, w):Q \in Sel(v)$$

A path selection  $(V, Sel)$  is *sound* iff it satisfies the following condition. For all  $v, P, Q, R$ , where  $P:Q:R \in Sel(v)$  and  $Q$  is a cycle,

$$P:R \in Sel(v)$$

### Theorem 1

A path selection  $(V, Sel)$  is satisfiable in network  $N, N = (V, E)$ , if each of the following conditions hold:

1.  $(V, Sel)$  is consistent
2.  $(V, Sel)$  is sound
3. For every  $v \in V$ , there exists a  $P, P \in Sel(v)$ , such that  $P$  is a network path in  $N$ .



### Corollary 1

Let  $(V, \text{Sel})$  and  $N$  be defined as in Theorem 1. Also, let each node  $v, v \in V$ , choose as its next hop to the root a neighbor  $w$ , where  $w$  is the first node of path  $P$ , and  $P$  is the shortest network path in  $\text{Sel}(v)$ . Then, this choice of next hops form a satisfiability tree  $T$  for  $(V, \text{Sel})$  in network  $N$ .



Theorem 1 states that if a path selection is consistent and sound, and each node has at least one selectable path, then a satisfiability tree may be found. Thus, all nodes may forward their data to the root along an acceptable path.

Since some path selections may not satisfy the optimality principle, nodes cannot simply pick as their next hop neighbor any neighbor along a selectable path. Corollary 1 states that if the neighbor chosen is any neighbor along the *shortest* selectable path, then a satisfiability tree is constructed.

### 3. Path Rankings

Above, we made no assumptions as to how a path selection  $(V, \text{Sel})$  is chosen. The selection may reflect policy routing decisions, such as restricting traffic across certain nodes. On the other hand, it may be desirable to define the selection with respect to some function that assigns ranks to paths.

The rank of a path may be based on a typical metric, such as bandwidth or delay, or it may be based on other factors. In particular, a ranking on paths based on a combination of a typical metric, such as bandwidth, and on policy based routing, is presented in the next section.

Each node desires its messages to be routed along the highest ranked path to the root, i.e., along a path that is optimal with respect to the ranking function. Thus, the path selection would correspond to those paths that are optimal with respect to the ranking. Hence, the path selection may change as the network topology changes.

More formally, a *path ranking* is a pair  $(V, \text{rank})$ , where  $\text{rank}$  is a function from rooted paths in  $V$  to non-negative integers.

Let  $P$  be a rooted path in network  $N, N = (V, E)$ .  $P$  is said to be *optimal* with respect to ranking  $(V, \text{rank})$  and network  $N$ , iff, for all rooted paths  $Q$  in  $N$ , where  $\text{head}(Q) = \text{head}(P)$ , we have  $\text{rank}(Q) \leq \text{rank}(P)$ .

A path selection  $(V, \text{Sel})$  is *optimal* with respect to network  $N, N = (V, E)$ , and path ranking  $(V, \text{rank})$ , iff, for all  $v \in V$ , and for all  $P \in \text{Sel}(v)$ ,  $P$  is optimal with respect to  $(V, \text{rank})$  and  $N$ .

Again, since an optimal path selection may not satisfy the optimality principle, a satisfiability tree may not exist for an optimal path selection. We next present sufficient properties to ensure that an optimal path selection is satisfiable.

A path ranking  $(V, \text{rank})$  is *consistent* iff it satisfies the following condition. For all  $v, w, P$ , and  $Q$ , where  $v = \text{head}(P), w \in P, w = \text{head}(Q)$ ,

$$\text{rank}(Q) \geq \text{rank}(P(w, \text{root})) \Rightarrow \text{rank}(P(v, w):Q) \geq \text{rank}(P)$$

A path ranking  $(V, \text{rank})$  is *sound* iff it satisfies the

following condition. For all  $v, P, Q, R$ , where  $P:Q:R \in \text{Sel}(v)$  and  $Q$  is a cycle,

$$\text{rank}(P:R) \geq \text{rank}(P:Q:R)$$

### Theorem 2

Let  $(V, \text{rank})$  be a consistent and sound path ranking, and let  $N = (V, E)$  be any network. Let  $(V, \text{Sel})$  be the optimal path selection with respect to  $(V, \text{rank})$  and  $N$ . Then,  $(V, \text{Sel})$  is satisfiable in  $N$ .



### Corollary 2

Let  $(V, \text{Sel})$  be defined as in Theorem 2. Let each node  $v, v \in V$ , choose as its next hop to the root a neighbor  $w$ , where  $w$  is the first node of the shortest path  $P$  in  $\text{Sel}(v)$ . Then, this selection of next hop neighbors forms a satisfiability tree  $T$  for  $(V, \text{Sel})$  in network  $N$ .



Theorem 2 states that if an optimal path selection is consistent and sound, then a satisfiability tree may be found. Thus, all nodes may forward their data to the root along an optimal path.

Since some optimal path selections may not satisfy the traditional optimality principle, nodes cannot simply pick as their next hop neighbor any neighbor along a selectable path. Corollary 2 states that if the neighbor chosen is any neighbor along the shortest optimum path, then a satisfiability tree is constructed.

### 4. Edge Metrics

In this section, we consider routing metrics, which are based solely on weight values assigned to the edges of the network. In [6], a general model encompassing a wide variety of edge metrics is introduced. This model encompasses typical metrics such as maximum bandwidth, minimum cost, and minimum delay metrics.

Necessary and sufficient conditions were also presented in [6] for edge routing metrics in this model to be free of conflicts. That is, a spanning tree always exists, regardless of network topology and weight assignment to each edge, which provides the best path from each node in the network to the destination node.

Below, we compare the model of edge metrics with our model of path ranking functions. We map each instance of edge routing metrics into an instance of path ranking function. We show that if the edge routing metric is free from conflicts, then the corresponding path ranking function is also free of conflicts, and thus, its optimal selection of paths is satisfiable.

Since path rank functions are more general than metrics based on edge weights, we present an example of a path ranking function that cannot be represented in the model of edge metrics. We begin by presenting the model in [6].

A *metric* is a five tuple  $(M, W, \text{met}, m_r, <)$  where:

- $M$  is a set of metric values
- $W$  is a set of edge weights
- $\text{met}$  is a metric function whose domain is  $M \times W$  and whose range is  $M$
- $m_r$  is the metric value assigned to the root node
- $<$  is an irreflexive, transitive and total binary relation over  $M$ .  $m \leq m'$  is defined as  $m = m' \vee m < m'$ .

A metric  $(M, W, \text{met}, \text{mr}, <)$  is assigned to a network  $N$ , by choosing a function  $\text{wf}$  that assigns to each edge  $(u, v)$  of  $N$  an edge weight  $\text{wf}(u, v)$  from  $W$ .

The metric of a rooted path  $P$  is obtained by successively applying the weight function starting at the root node and proceeding downwards along  $P$ . That is, if  $P = v_k, v_{k-1}, \dots, v_0$ , where  $v_0 = \text{root}$ , then the metric value of  $P$  is  $m_k$ , where:

- $m_0 = m_r$
- for each  $i, k \geq i > 0, m_i = \text{mf}(m_{i-1}, \text{wf}(v_i, v_{i-1}))$

The objective is to obtain a spanning tree  $T$  over a network  $N$ , such that for each node  $v$ , the rooted path along  $T$  from  $v$  to the root is a maximum metric path. That is, there is no other path from  $v$  to the root with a greater metric.

The following metric properties are presented in [6].

A metric  $(M, W, \text{met}, \text{mr}, <)$  is *monotonic* iff for every pair of metric values  $m$  and  $m'$ , and for every edge weight value  $w$ , if  $m \leq m'$  then  $\text{mf}(m, w) \leq \text{mf}(m', w)$ . From monotonicity it follows that  $\text{metric}(Q) \geq \text{metric}(R) \Rightarrow \text{metric}(P:Q) \geq \text{metric}(P:R)$  for all  $P, Q$ , and  $R$ .

A metric  $(M, W, \text{met}, \text{mr}, <)$  is *bounded* iff for every metric value  $m$  and edge weight value  $w$ ,  $m \leq \text{mf}(m, w)$ . From boundedness, it follows that  $\text{metric}(Q) \geq \text{metric}(P:Q)$  for all  $P$  and  $Q$ .

In [6], it was shown that a metric is both monotonic and bounded iff for any network  $N$  and any weight function  $\text{wf}$ , there exists a spanning tree  $T$  that provides the maximum metric to each node. We next show that these requirements on the metric imply our sufficient condition for optimal paths.

### Theorem 3

Let  $(M, W, \text{met}, \text{mr}, <)$  be a monotonic and bounded metric assigned to network  $N, N = (V, E)$ . Let  $\Phi$  be a function from  $M$  to the positive integers, such that, for all  $m$  and  $m'$  in  $M, m < m'$  iff  $\Phi(m) < \Phi(m')$ . This is possible since the set  $M$  is totally ordered. For all rooted paths  $P$  in  $V$ , let  $\text{rank}(P) = \Phi(\text{metric}(P))$ . Then:

- a)  $(V, \text{rank})$  is consistent and sound.
- b) Let  $(V, \text{Sel})$  be the optimum path selection with respect to  $(V, \text{rank})$  and  $N$ . Then,  $(V, \text{Sel})$  is satisfiable in  $N$ .

◆

From the definition of  $\Phi$ , a maximum metric path in the network is also an optimum path with respect to the path ranking function and network  $N$ . From Theorem 3, the requirements in [6] for a maximizable metric imply our requirements to satisfy the corresponding optimum path selection. Thus, the model of path ranking functions subsumes the model of routing metrics based on edge weights.

Next, we finalize this section by presenting a simple example of a path ranking function whose optimal path selection is satisfiable, but does not satisfy the requirement of boundedness required for edge routing metrics.

Let the set of nodes  $V$  be divided into two disjoint sets,  $L$  (leaf nodes) and  $S$  (service providers). For each  $v \in L$ , there is a single node  $v' \in S$ , known as the *provider* of  $v$ .

Each edge is assigned a weight in the range  $1 \dots N-1$ , which denotes the bandwidth of the edge. The rank of a

path  $P$  is defined as follows:

- a) if there is an edge  $(u, v)$  in  $P$ , where  $v \in L$ , then  $\text{rank}(P) = 0$ .
- b) if (a) does not apply, and the first edge of  $P$  is  $(v, v')$ , where  $v \in L, v' \in S$ , and  $v'$  is the provider of  $v$ , then  $\text{rank}(P) = N$ .
- c) if (a) and (b) do not apply, then the  $\text{rank}(P)$  is simply the bottleneck bandwidth of  $P$ .

These rules represent the following scenario. Leaf nodes in  $L$  do not accept traffic through them unless it is absolutely necessary (i.e., unless the neighboring node has no other choice but to forward its data through this node). This is the motivation for rule (a). Also, each leaf node has a preferred provider, and it will always desire to forward its data to this provider. This is the motivation for rule (b). Finally, if the path is not going through nor originating at a leaf node, then the rank of the path is simply its bottleneck bandwidth (rule (c)).

Note that if we consider the rank of a path as the metric of the path, then this metric does not satisfy the boundedness property. To see this, consider a rooted path whose first node  $v$  is a node in  $S$ . Let its rank be  $x$  ( $x \leq N-1$ ). If we extend this path to a node  $u, u \in L$  and  $v$  being the provider of  $v$ , then the rank of the path is  $N$ , i.e., the rank increases, which violates the boundedness property. Thus, by the results in [6] it is not maximizable.

However, it is easy to show that this rank function is both consistent and sound, and thus, its optimal selection of paths is satisfiable. This appears to contradict the results presented in [6]. However, there is no contradiction, because the model of edge routing metrics does not capture the entire meaning of rules (a) through (c) above, since it cannot take node identities into consideration. Thus, the results in [6] do not apply to this extended model.

## 5. Convergence of Optimum Routing

In the previous sections, we have shown necessary properties for a path ranking to have a satisfiable set of optimum paths. In this section, we present a distributed algorithm to compute the optimum path from every node to the destination. The only assumption made is that the path ranking function used is consistent and sound.

The algorithm is based on the periodic exchange of paths with neighboring nodes, as done in [11]. Each node  $v$  maintains a variable  $P$ , where it stores its desired path to the root. Each node  $v$  periodically sends its path  $P$  to its neighbors. Whenever a neighbor offers a better path  $Q$  to the destination, node  $v$  adopts  $Q$  instead.

The algorithm for a non-root node  $v$  follows.

```

node v
variables
  P : sequence of nodes { path to the root node }
  Q : sequence of nodes { path rcvd. from neighbors }
repeat
  if for every neighbor u of v, # update in channel from v to u = 0
  then
    sanity(P);
    for each neighbor u of v,
      send update(v:P) to u
    end for
  end if
  if receive update(Q) from any neighbor u then

```

```

sanity(Q);
if head(P) = u ∨ P = ∅ ∨ rank(Q) < rank(P) ∨
(rank(Q) = rank(P) ∧ |Q| < |P|)
then
    P := Q; sanity(P);
end if
end if
end repeat

```

Above,  $\text{sanity}(P)$  is a shorthand for the following:

```

if (for all neighbors u, head(P) ≠ u) ∨
tail(P) ≠ root ∨ cycle(P) then P := ∅
end if

```

$\text{Sanity}(P)$  is used to ensure that path  $P$  is a sensible path, i.e., it has no cycles, its first node is a neighbor, and the last node is the root. Periodically, node  $v$  forwards a copy of  $P$  to all its neighbors, provided the previous copy has already been received. Furthermore, if node  $v$  receives a path  $Q$  from its neighbor  $u$ , and it offers a better path than  $P$ , then  $v$  adopts  $Q$  as its new path. The algorithm for the root node follows.

```

node root
repeat
if (for every neighbor u, # update in channel from root to u = 0)
then
    for each neighbor u of v do
        send update(root) to u
    end for
end if
end

```

We next consider the correctness of the protocol.

#### Lemma 1

Let  $(V, \text{rank})$  be a consistent and sound path ranking. There exists an alternative ranking,  $(V, \text{rank}')$ , satisfying the following properties.

1. for all rooted paths  $P$  and  $Q$  with a common initial node,  $\text{rank}(P) < \text{rank}(Q) \Rightarrow \text{rank}'(P) < \text{rank}'(Q)$
2. for all rooted paths  $P$  and  $Q$  with a common initial node,  $\text{rank}(P) = \text{rank}(Q) \Rightarrow \text{rank}'(P) = \text{rank}'(Q)$
3. for all rooted paths  $P$ , where  $v$  is the first node of  $P$ , and all nodes  $w$ ,  $w \notin P$ ,  $\text{rank}'(P) \geq \text{rank}'(P:(v,w))$

◆ Lemma 1 simply states that if there is a path ranking which is consistent and sound, then there exists another path ranking which orders paths in the same order as the original, but in addition, the rank of a path cannot increase if it is extended to include additional nodes. We use this lemma to prove the following theorem.

#### Theorem 4

Let  $(V, \text{rank})$  be a consistent and sound path ranking. Assume all nodes of a network  $N$ ,  $N = (V, E)$ , execute the above algorithm. Then,

- a) The algorithm reaches a steady state, i.e., variable  $P$  in all nodes ceases to change. Also,
- b) Let  $(V, \text{Sel})$  be the optimal path selection, i.e., for all  $v \in V$ ,  $\text{Sel}(v)$  is optimal with respect to  $(V, \text{rank})$  and  $N$ . Then, at a steady state, the union of all  $P$  variables of all nodes form a satisfiability tree for  $(V, \text{Sel})$ .

◆ Theorem 4 states that the above algorithm will indeed terminate, and furthermore, the union of all  $P$  variables of

all nodes form a spanning tree, and each path  $P$  is an optimal path. Thus, if each node chooses  $\text{head}(P)$  as the next hop neighbor, its data messages are guaranteed to follow an optimum path to the root.

## 6. Concluding Remarks

In [7][8], an abstraction of the BGP problem, known as path vector routing, is introduced. Each node has a set of paths to the destination which is willing to accept. This set is totally ordered. One problem with this protocol is that it may oscillate, and never establish a path to the destination. In [7] sufficient conditions are given to guarantee convergence.

The difference between this work and our model is that the conditions presented in [7] allow the system to converge to a path which is not optimum for every node. Actually, it may converge to a solution where some nodes obtain no path to the destination. Our work is mostly concerned with ensuring that each node does indeed obtain an optimum path.

In [9] a modified algorithm is presented which guarantees convergence. It does so by removing from the set of acceptable paths of a node those paths which at run time are detected to be involved in a race condition.

The problem with the algorithm presented in [9] is that the removed path may later become necessary as the network topology changes. In our future work, we will present an algorithm which guarantees convergence, but without removing any paths from the acceptable set. The algorithm will attempt at first to follow the path ordering originally given, but if it fails to produce a result it will switch to an alternative ranking built at run time which guarantees convergence.

## References

- [1] R. E. Bellman, *Dynamic Prog.*, Princeton Univ. Press, 1957.
- [2] S. Chen, K. Nahrstedt, "An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions", *IEEE Network*, November/December 1998.
- [3] J. Cobb, M. Waris, "Propagated Timestamps: A Scheme for the Stabilization of Maximum-Flow Routing Protocols", *Proc. of the Third Workshop on Self-Stabilizing Systems*, 1997
- [4] E. W. Dijkstra, "A Note on Two Problems on Connection with Graphs", *Numerical Mathematics*, Vol. 1, 1959.
- [5] Gouda, M. G. and M. Schneider, "Maximum Flow Routing", *Proceedings of the Second Workshop on Self-Stabilizing Systems*, Technical Report, Department of Computer Science, University of Nevada, Las Vegas, May 1995
- [6] Gouda, M. G. and M. Schneider, "Maximizable Routing Metrics", *Proceedings of the IEEE International Conference on Network Protocols*, 1998.
- [7] T. Griffin, F. B. Shepherd, G. Wilfong, "Policy Disputes in Path-Vector Protocols", *Proceedings of the IEEE International Conference on Network Protocols*, 1999.
- [8] T. Griffin, G. Wilfong, "An Analysis of BGP Convergence Properties", *Proc. of the ACM SIGCOM Conference*, 1999.
- [9] T. Griffin, G. Wilfong, "A Safe Path Vector Protocol", *Proceedings of the IEEE INFOCOM Conference*, 2000.
- [10] Y. Rekhter and T. Li, "A Border Gateway Protocol", RFC 1771 (BGP Version 4), 1995.
- [11] K. Shin, M. Chen, "Performance Analysis of Distributed Routing Strategies Free of Ping-Pong-Type Looping", *IEEE Transactions on Computers*, 1987.
- [12] A. S. Tanenbaum, *Computer Networks*, 3<sup>rd</sup> edition, Prentice-Hall Inc., 1996.
- [13] Z. Wang, J. Crowcroft, "Bandwidth-Delay Based Routing

Algorithm", *Proceedings of the IEEE Global Telecommunications Conference, 1995*

## Appendix - Proofs of Theorems

### Theorem 1

A path selection  $(V, Sel)$  is satisfiable in network  $N$ ,  $N = (V, E)$  if all the following conditions hold:

1.  $(V, Sel)$  is consistent
2.  $(V, Sel)$  is sound
3. For every  $v \in V$ , there exists a  $P$ ,  $P \in Sel(v)$ , such that  $P$  is a network path in  $N$ .

*Proof*

We prove the theorem by exhibiting an algorithm which constructs a satisfiability tree  $T$  under the assumptions of the theorem. The algorithm follows.

$T$  : set of edges {initially empty}

$P$  : sequence of edges

$u$  : node

$v$  : node

```

while nodes(T) ≠ V do
  u := any node not in T;
  P := a shortest satisfiable path in Sel(u);
  v := u;
  while v ≠ root ∧ v not in T do
    w := next node after v in P;
    if P(w, root) is a shortest satisfiable path of w then
      v := w;
    else
      Q := a shortest satisfiable path in Sel(w);
      P := P(u, w):Q ;
      v := w;
    end if
  end while
  T := T ∪ P(u, v);
end while

```

At all times in the algorithm,  $T$  is a tree subgraph of  $N$  and

- (i)  $(\forall x, x \in T, T(x, \text{root}) \text{ is satisfiable})$

The algorithm augments  $T$  as follows. Any node  $u$  not in  $T$  is chosen, and its shortest satisfiable path  $P$  in  $Sel(u)$  is chosen. From the assumptions of the theorem, such path must exist. The algorithm then iterates over the nodes of  $P$ , starting from  $u$  and moving towards the root, as illustrated in Figure 2. At all times, the algorithm ensures the following,

- (ii)  $(\forall x, x \in P(u, v), P(x, \text{root}) \text{ is satisfiable})$

where  $v$  is the iteration variable. We next show how (ii) is preserved.

Consider the node  $w$  after  $v$  in  $P$ . We have two cases to consider for  $w$ .

- a)  $P(w, \text{root})$  is a shortest satisfiable path of  $w$ . In this case, (ii) above still holds if we increase the range of  $x$  to include  $w$ , so the algorithm assigns  $w$  to  $v$ .
- b)  $P(w, \text{root})$  is not a shortest satisfiable path of  $w$ . In this case, a shortest satisfiable path, namely,  $Q$ , is chosen. From the assumptions of the theorem, such path must exist. Note that for all  $x, x \in P(u, v)$ ,  $P(x, w):Q$  is satisfiable. This is because  $(V, Sel)$  is consistent and  $P(x, \text{root})$  is satisfiable. Therefore, we update  $P$  to  $P(u, w):Q$ , and  $v$  is set to  $w$ . Thus, (ii) above continues to hold.

However, in b) above,  $Q$  must not form a loop with  $P$ , i.e., no node in  $Q$  is a node in  $P(u, v)$ . This cannot occur for the following reason.

Let  $Q$  contain a node  $x, x \in P(u, v)$ , as shown in Figure 2(b). From the consistency property and (ii),  $P(x, w):Q$  is satisfiable. Thus, from the soundness property,  $Q(x, \text{root})$  is satisfiable.

Let  $y$  be the next node after  $x$  in  $P$  ( $y = v$  in Figure 2(b)). Let  $R$  be the shortest satisfiable path of  $x$ , where  $y$  is the second node in  $R$  (such an  $R$  exists from the previous iterations of the algorithm). Since  $x$  chose  $y$  as its next hop,  $|R| \leq |Q(x, \text{root})|$ . From the consistency property,  $Q(w, x):R$  is satisfiable. Hence, since  $|R| \leq |Q(x, \text{root})|$ ,  $w$  may choose  $Q(w, x):R$  as its shortest satisfiable path. This path goes through node  $y$ .

By applying the same argument to  $y$  as we did to  $x$ ,  $w$  may choose  $Q(w, y):S$  as its shortest satisfiable path, where  $S$  is the shortest satisfiable path of  $y$ , and so on, until we conclude that  $Q(w, x):P(x, w):U$  is a shortest satisfiable path of  $w$ , where  $U$  is a shortest satisfiable path of  $w$ . This is a contradiction, since  $U$  and  $Q(w, x):P(x, w):U$  cannot both be shortest satisfiable paths. Thus,  $Q$  cannot form a loop.

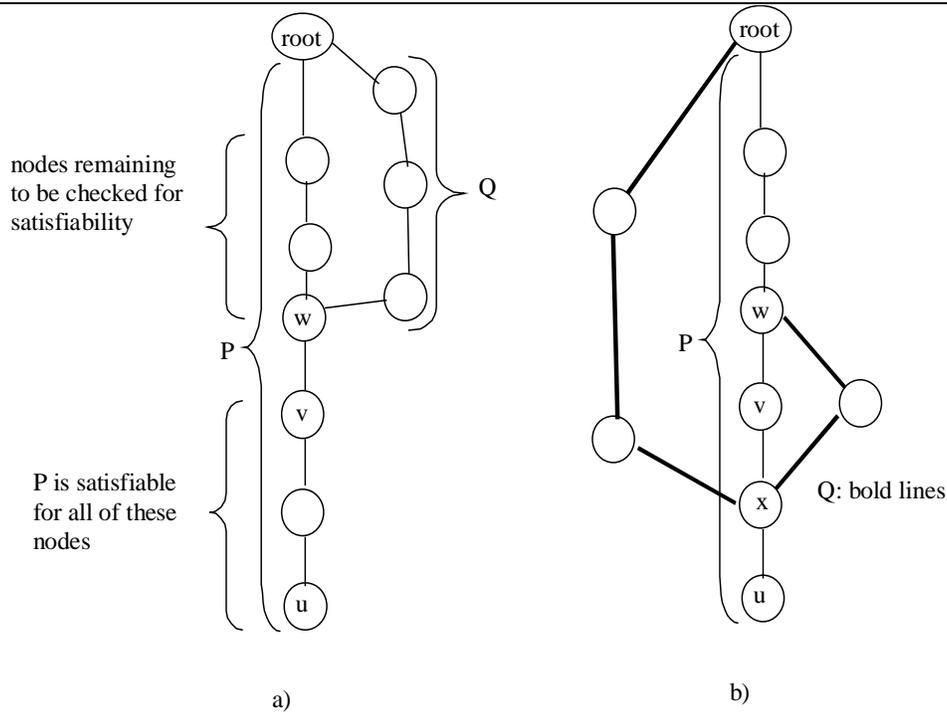


Figure 2: Path traversal by the algorithm

The traversal of path  $P$  ends when  $v = \text{root} \vee v \in T$ . Then,  $P(u, v)$  is added to  $T$ . If  $v = \text{root}$ , then no node in  $P$ , other than the root, is in  $T$ . From (ii), we can add  $P$  to  $T$  without violating (i). If  $v \in T$ , then, consider any  $x$  in  $P(u, v)$ . From (ii),  $P(x, v):P(v, \text{root}) \in \text{Sel}(x)$ . Also, from (i),  $T(v, \text{root}) \in \text{Sel}(v)$ . Hence, from consistency,  $P(x, v):T(v, \text{root}) \in \text{Sel}(x)$ . Thus, we can add  $P(u, v)$  to  $T$  without violating (i).

The algorithm ends when  $T$  is a spanning tree of  $N$ , which from (i), we conclude it is a satisfiability tree.



**Theorem 2**

Let  $(V, \text{rank})$  be a consistent and sound path ranking, and let  $N = (V, E)$  be any network. Let  $\text{Sel}(v)$  be the set of optimal paths in  $N$  from  $v$  to the root. Then,  $(V, \text{Sel})$  is satisfiable in  $N$ .

*Proof*

We show below that  $(V, \text{Sel})$  is both consistent and sound. Since all paths in  $(V, \text{Sel})$  are in  $N$  (by definition of optimal paths), then the conditions of Theorem 1 hold, and  $\text{Sel}(v)$  is satisfiable in  $N$ .

Consider first the consistency of  $(V, \text{Sel})$ . Let  $P \in \text{Sel}(v)$ ,  $w \in P$ , and  $Q \in \text{Sel}(w)$ . We are required to show that  $P(v, w):Q \in \text{Sel}(v)$ .

Since  $Q \in \text{Sel}(w)$ ,  $Q$  is an optimal path for  $w$ , thus,  $\text{rank}(Q) \geq \text{rank}(P(w, \text{root}))$ .

From the consistency of the rank,  $\text{rank}(P(v, w):Q) \geq \text{rank}(P(v, w):P(w, \text{root})) = \text{rank}(P)$ .

Since  $P \in \text{Sel}(v)$ ,  $P$  is an optimal path for  $v$ , thus  $\text{rank}(P) \geq \text{rank}(P(v, w):Q)$ .

Thus,  $\text{rank}(P) = \text{rank}(P(v, w):Q)$ , and since  $P$  is optimal for  $v$ , then  $P(v, w):Q$  is also optimal for  $v$ , and  $P(v, w):Q \in \text{Sel}(v)$ .

Consider now the soundness property. For all  $v, P, Q, R$ , where  $P:Q:R \in \text{Sel}(v)$  and  $Q$  is a cycle, we are required to show  $P:R \in \text{Sel}(v)$ .

From soundness of the rank,  $\text{rank}(P:R) \geq \text{rank}(P:Q:R)$ .

From  $P:Q:R \in \text{Sel}(v)$ ,  $P:R$  is optimal, thus  $\text{rank}(P:Q:R) \geq \text{rank}(P:R)$ .

Hence,  $\text{rank}(P:Q:R) = \text{rank}(P:R)$ , so  $P:R$  is optimal, and  $P:R \in \text{Sel}(v)$ .



**Corollary 2**

Let  $(V, \text{Sel})$  be defined as in Theorem 2. Let each node  $v, v \in V$ , choose as its next hop to the root a neighbor  $w$ , where  $w$  is the first node of the shortest path  $P$  in  $\text{Sel}(v)$ . Then, this selection of next hop neighbors form a satisfiability tree  $T$  for  $(V, \text{Sel})$  in network  $N$ .

*Proof*

From the proof of Theorem 2,  $(V, \text{Sel})$  is sound and consistent, and from the definition of  $(V, \text{Sel})$ , every path in  $\text{Sel}(v)$  is in network  $N$ . Thus, the algorithm used in the proof of Theorem 1 may be used to find the satisfiability tree  $T$ . In the construction

of  $T$ , each node  $v$  chooses its neighbor  $w$  along a shortest path in  $\text{Sel}(v)$  as the next hop to the root, as desired.

◆

### Theorem 3

Let  $(M, W, \text{met}, m_r, <)$  be a monotonic and bounded metric assigned to network  $N$ ,  $N = (V, E)$ . Let  $\Phi$  be a function from  $M$  to the positive integers, such that, for all  $m$  and  $m'$  in  $M$ ,  $m < m'$  iff  $\Phi(m) < \Phi(m')$ . This is possible since the set  $M$  is totally ordered. For all rooted paths  $P$  in  $V$ , let  $\text{rank}(P) = \Phi(\text{metric}(P))$ . Then:

a)  $(V, \text{rank})$  is consistent and sound.

b) Let  $(V, \text{Sel})$  be the optimum path selection with respect to  $(V, \text{rank})$  and  $N$ . Then,  $(V, \text{Sel})$  is satisfiable in  $N$ .

*Proof*

Note that since  $<$  is a total, irreflexive, and transitive relation on the metric values, a one to one mapping function,  $\text{map}$ , from metrics to non-negative integers, may be obtained such that, for every  $m_1, m_2 \in M$ ,  $m_1 < m_2$  iff  $\text{map}(m_1) < \text{map}(m_2)$ . Thus, we define  $\text{rank}(P) = \text{map}(\text{metric}(P))$  for all rooted paths  $P$ .

Consider first consistency. Let  $P \in \text{Sel}(v)$ ,  $w \in P$ , and  $Q \in \text{Sel}(w)$ .

$$\begin{aligned} & \text{rank}(Q) \geq \text{rank}(P(w, \text{root})) \\ \Leftrightarrow & \{ \text{from the definition of rank} \} \\ & \text{metric}(Q) \geq \text{metric}(P(w, \text{root})) \\ \Rightarrow & \{ \text{from monotonicity of metric} \} \\ & \text{metric}(P(v, w): Q) \geq \text{rank}(P) \\ \Leftrightarrow & \{ \text{from the definition of rank} \} \\ & \text{rank}(P(v, w): Q) \geq \text{rank}(P) \end{aligned}$$

Thus,  $\text{rank}(Q) \geq \text{rank}(P(w, \text{root})) \Rightarrow \text{rank}(P(v, w): Q) \geq \text{rank}(P)$ , and  $(V, \text{rank})$  is consistent.

Consider now boundedness. Let  $P = Q:R:S$  where  $x$  is the first and last node of  $R$ .

$$\begin{aligned} & \text{true} \\ = & \{ \text{from metric boundedness} \} \\ & \text{metric}(R:S) \leq \text{metric}(S) \\ \Rightarrow & \{ \text{from metric monotonicity} \} \\ & \text{metric}(Q:R:S) \leq \text{metric}(Q:S). \\ \Leftrightarrow & \{ \text{from the definition of rank} \} \\ & \text{rank}(Q:R:S) \leq \text{rank}(Q:S) \end{aligned}$$

Thus,  $\text{rank}(Q:R:S) \leq \text{rank}(Q:S)$  holds, and  $(V, \text{rank})$  is sound.

◆

### Lemma 1

Let  $(V, \text{rank})$  be a consistent and sound path ranking. There exists an alternative ranking,  $(V, \text{rank}')$ , satisfying the following properties.

1. for all rooted paths  $P$  and  $Q$  with a common initial node,  $\text{rank}(P) < \text{rank}(Q) \Rightarrow \text{rank}'(P) < \text{rank}'(Q)$
2. for all rooted paths  $P$  and  $Q$  with a common initial node,  $\text{rank}(P) = \text{rank}(Q) \Rightarrow \text{rank}'(P) = \text{rank}'(Q)$
3. for all rooted paths  $P$ , where  $v$  is the first node of  $P$ , and all nodes  $w$ ,  $w \notin P$ ,  $\text{rank}'(P) \geq \text{rank}'(P:(v, w))$

*Proof*

Consider a directed graph  $G$ ,  $G = (V', E')$ , where:

- the nodes in  $V'$  are the rooted paths of  $V$ .
- $E'$  consists only of the following two types of edges:
  - a) **edges between paths sharing the same first node:** for any two nodes  $P$  and  $Q$  in  $V'$ , if  $\text{head}(P) = \text{head}(Q)$  and  $\text{rank}(P) \leq \text{rank}(Q)$ , then  $(P, Q) \in E'$ . Note that if  $\text{head}(P) = \text{head}(Q)$  and  $\text{rank}(P) = \text{rank}(Q)$  then both  $(P, Q)$  and  $(Q, P)$  will exist in  $E'$ .
  - b) **edges between a path and its extension by one node:** for every pair node  $w \in V$  and path  $P$  of nodes in  $V$ ,  $(w:P, P) \in E'$ .

We next argue that there is no cycle in  $G$ , where  $(P, Q)$  is an edge in  $G$  and  $\text{rank}(P) < \text{rank}(Q)$ .

The cycle in  $G$  cannot contain only edges of type (a) above. This is because the function  $\text{rank}$  returns integers, and a cycle would violate the total ordering property of the set of integers.

Therefore, we can group consecutive edges in the cycle in  $G$  according to their category above, (a) or (b). For example, in Figure 3, edges of type (a) are solid, while edges of type (b) are dashed. Notice that all the nodes of consecutive edges of type (a), by definition of (a), have the same starting node. Hence,  $P_0, A, B$ , and  $R_k$  all have the same starting node in  $V$ . Let this node be  $x$ . Similarly,  $P_i$  and  $Q_0$  have the same starting node,  $y$ , and  $Q_j$  and  $R_0$  have the same starting node,  $z$ .

Also,  $P_0, P_1, \dots, P_i$ , are increasing extensions of  $P_0$ . I.e.,  $P_1 = w_0:P_0$  for some  $w_0 \in V$ ,  $P_2 = w_1:P_1$  for some  $w_1 \in V$ , etc.. Similarly,  $Q_0, Q_1, \dots, Q_j$  are increasing extensions of  $Q_0$ , and  $R_0, R_1, \dots$ , are increasing extensions of  $R_0$ .

Consider an edge among those whose paths begin in node  $x$ , e.g., edge  $(A, B)$ . Thus  $\text{rank}(A) \leq \text{rank}(B)$ . Assume that

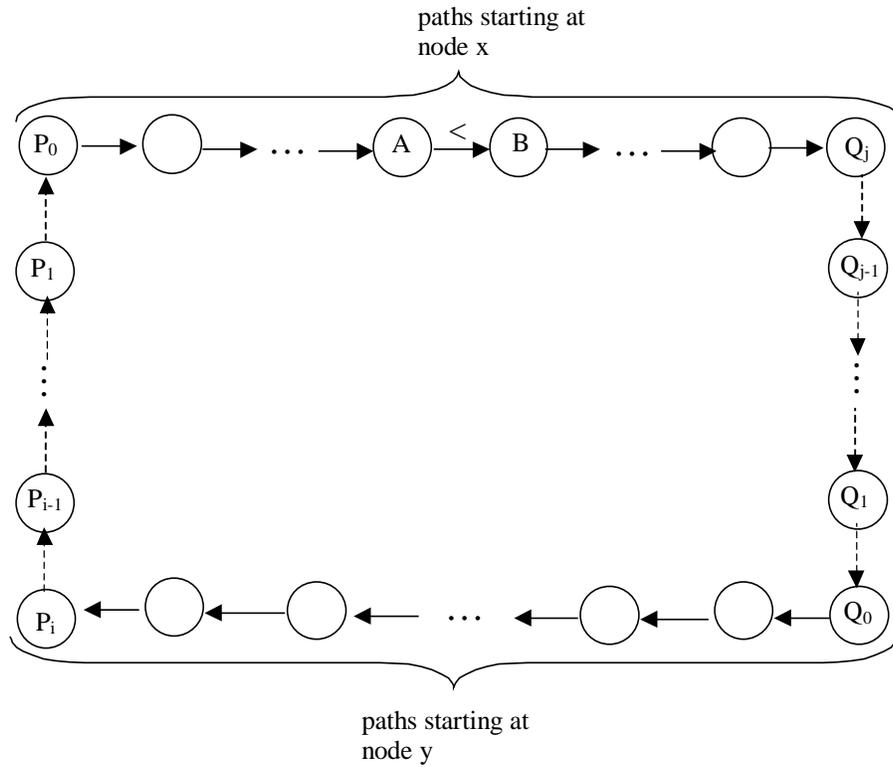


Figure 3

$\text{rank}(A) < \text{rank}(B)$ . We must show this is not possible.

From the definition of (a) edges,  $\text{rank}(P_0) \leq \text{rank}(A)$ .

Extend path A using the same nodes to extend  $P_0$  into  $P_i$ . Call this new path  $A_y$  (i.e., "A extended until node y"). From the consistency of the rank,  $\text{rank}(P_i) \leq \text{rank}(A_y)$ .

From the definition of (a) edges,  $\text{rank}(Q_0) \leq \text{rank}(P_i)$ , and hence,  $\text{rank}(Q_0) \leq \text{rank}(A_y)$ .

Extend now  $A_y$  using the same nodes to extend  $Q_0$  into  $Q_j$ . Call this new path  $A_z$  (i.e., "A extended until node z"). From the consistency of the rank,  $\text{rank}(Q_j) \leq \text{rank}(A_z)$ .

From the definition of (a) edges,  $\text{rank}(R_0) \leq \text{rank}(Q_j)$ , and hence,  $\text{rank}(R_0) \leq \text{rank}(A_z)$ .

Extend now  $A_z$  using the same nodes to extend  $R_0$  into  $R_k$ . Call this new path  $A_x$  (i.e., "A extended until node x reached again"). From the consistency of the rank,  $\text{rank}(R_k) \leq \text{rank}(A_x)$ . Since the rank function is sound,  $\text{rank}(A_x) \leq \text{rank}(A)$ , and hence  $\text{rank}(R_k) \leq \text{rank}(A_x) \leq \text{rank}(A)$ . However, from the definition of (a) edges,  $\text{rank}(R_k) > \text{rank}(A)$ . Hence, a cycle is impossible.

This argument can be extended to any number of groups of consecutive edges of type (a), or reduced to the case of only a single one (i.e., the case where  $P_i$  also has x as its first node, and there is no node y or z in the cycle). The argument is the same.

Thus, since there is no cycle in the graph where one of the edges (A, B) has  $\text{rank}(A) < \text{rank}(B)$ , then we may assign the same value of  $\text{rank}'$  to all nodes in the cycle, without violating requirements 1 and 2 of the lemma.

Therefore,  $\text{rank}'$  is defined as follows. Consider the strongly connected components of G, and reduce each strongly connected component to a single node, obtaining a new graph  $G'$ . Perform a topological sort of the graph, and assign an integer to each strongly connected component according to this sort.  $\text{rank}'(P)$  for any path P in V is the integer of the topological sort corresponding to the strongly connected component of G containing P.

◆

**Theorem 4**

Let  $(V, \text{rank})$  be a consistent and sound path ranking. Assume all nodes of a network N,  $N = (V, E)$ , execute the above simple path finding algorithm. Then,

- a) The algorithm reaches a steady state, i.e., variable P in all nodes ceases to change. Also,
- b) Let  $(V, \text{Sel})$  be the optimal path selection, i.e., for all  $v \in V$ ,  $\text{Sel}(v)$  is optimal with respect to  $(V, \text{rank})$  and N. Then, at a steady state, the union of all P variables of all nodes form a satisfiability tree for  $(V, \text{Sel})$ .

*Proof*

First, notice that, in a system which exchanges paths to the root, regardless of the initial state of the system, eventually each P

variable in the network will contain either the empty sequence or a path in network  $N^2$ . Thus, we assume we have reached a state in which each  $P$  variable is a network path and this continues to hold forever.

From Lemma 1, we can replace the original rank function by a rank' function which does not affect the behavior of the protocol. However, for each path  $P$ ,  $\text{rank}'(P) \geq \text{rank}'(w:P)$  for any  $w$ , i.e., extending the path cannot increase its rank. Thus, we assume we are using rank' instead of rank.

With a slight abuse of notation, we denote by  $\text{rank}(v)$ ,  $v \in V$ , to be the rank' of an optimum rooted path from  $v$  to the root, and we denote by  $\text{length}(v)$  to be the length of this optimum path.

We construct a satisfiability tree  $T$  by starting from an initial tree  $T'$ , which contains only the root node, and expand  $T'$  until it becomes a spanning tree. For each node  $v$  in  $T'$ , the rooted path along  $T'$  is optimum for  $v$ . Furthermore, an edge added to  $T'$  is never removed.

Consider the pairs of values  $(\text{rank}'(v), \text{length}(v))$ . Order these pairs in lexicographical ordering (i.e., order them by rank' value, breaking ties in favor of length). We build  $T'$  in increasing order of this pair. I.e., at all times, for any node  $v$  in  $T'$  and any node  $w$  not in  $T'$ ,  $\text{rank}'(v) \geq \text{rank}'(w)$ .

Consider the nodes neighboring the root. Let  $v$  be the neighbor of root for which the rank of  $v:\text{root}$  is minimum. Thus, from the definition of rank', no path in the network can have a rank' smaller than  $v:\text{root}$ , and, since the length of the path is one, no shorter path exists. Thus, when  $v$  receives this path from the root, it will adopt it as its path. Notice that any other rooted path of  $v$ , even of the same rank' value, must have a length of at least two. This forces  $v$  to take path  $v:\text{root}$  as its path and keep it forever, and all nodes  $w$  other than  $v$  have  $(\text{rank}'(w), \text{length}(w)) \leq (\text{rank}'(v), \text{length}(v))$ .

Consider now the general case when  $T'$  has any number of nodes. Consider the node  $w$  outside of  $T'$ ; which has highest  $(\text{rank}'(w), \text{length}(w))$  pair. This node must be adjacent to the nodes in  $T'$ ; since otherwise, the next hop along the optimum path of  $w$  would have either a better rank than  $w$  or equal rank but shorter length. Let  $v$  be the node of  $T'$  adjacent to  $w$  and along the optimum path of  $w$ . Since  $v$  is in  $T'$ , its path will never change, and it will continue to send this path to  $w$ . Since there is no better path that  $w$  can have, eventually  $w$  chooses the optimum path through  $v$ , and continues to have this path forever.

Thus, eventually,  $T'$  is a spanning tree, it is stable, and all its rooted paths are optimal.

◆

---

<sup>2</sup> For a proof, see C. Alaettinoglu and U. Shankar, "Stepwise Assertional Design of Distance-Vector Routing Protocols", in *Protocol Specification, Testing, and Verification (PSTV) Symposium*, 1993.